



# DaHyun, Kim

## 0. Intro

### 1. Facial Verification on Image

- Face Detection
- Crop / Rotation
- Recognition / Determination

### 2. Facial Identification on Image

- Enroll / Gallery
- Facial Enroll
- Identification / Determination

### 3. Open Dataset & Metrics

- Dataset
- Verification result Analysis
- Identification result Analysis

### 1. Face Verification

- Face Verification - Manual
- Face Verification - use DeepFace.verify()

### 2. Face Recognition

- Create Gallery & Enroll
- Print all Gallery List
- Face Recognition
- Delete
- Clear



# Intro.

# 안면인식 기술 정의

## PIAI Research Department

- 안면인식(Facial Recognition, FR) : 얼굴 이미지를 통해 그 사람이 누구인지 (or 비교군의 동일인물인지) 판별하는 기술
- 안면인식의 경우 intra-class variation과 inter-class similarity 특징이 있음.
  - Intra-class variation : 같은 class이지만 전혀 달라 보일 수 있음 (variation에 취약)
  - Inter-class similarity : 다른 class이지만 매우 비슷해 보일 수 있음(similarity에 취약)

Intra-class  
variation

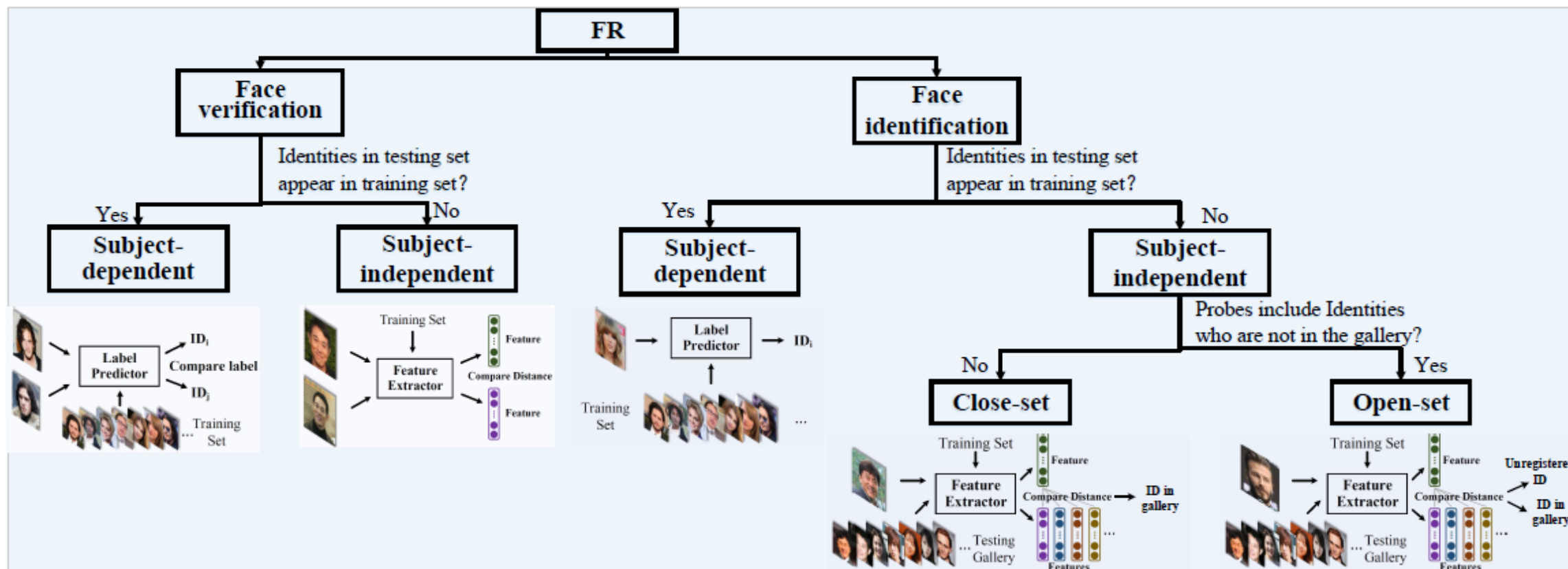


Inter-class  
variation



Marilyn Monroe

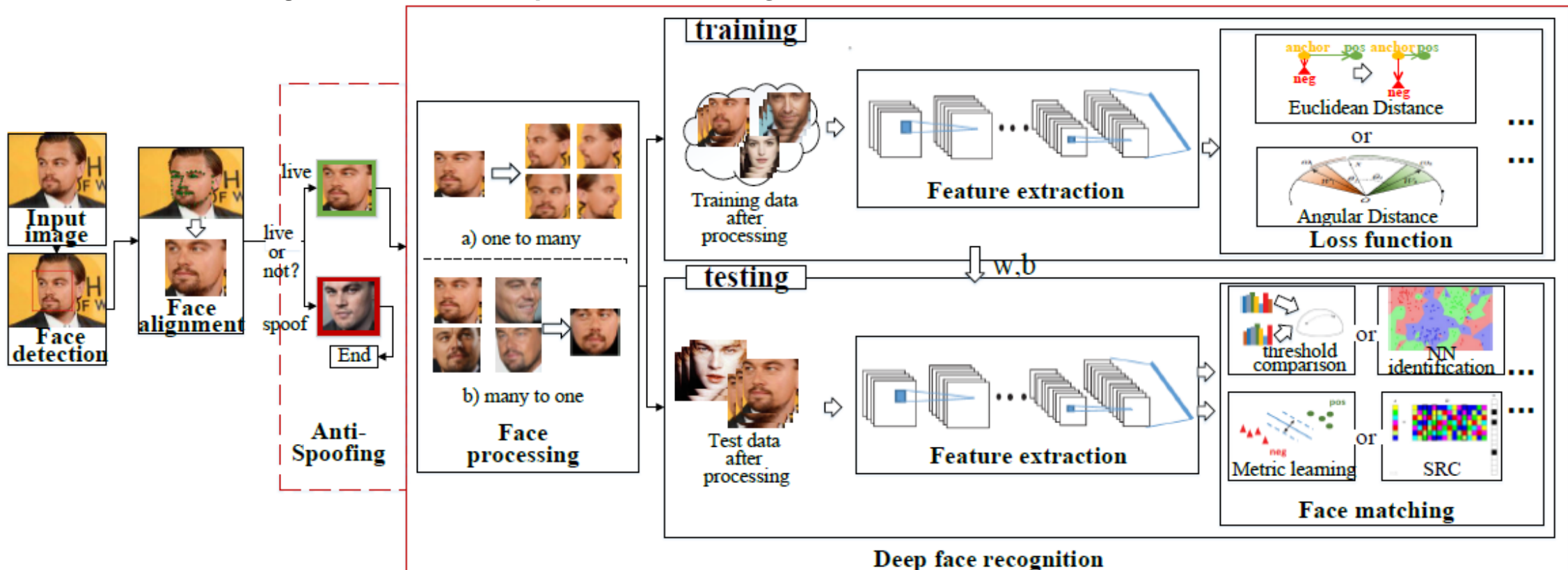
- Face verification : 기준 이미지와 비교 이미지의 사람이 동일 인물인가?
- Face identification(=Recognition) : 기준 이미지는 누구인가? (갤러리 : 인식할 사람-ID의 집합)
- Subject-dependent/independent : 평가 이미지가 학습에 쓰였는가?
- Close-set/Open-set : 기준 이미지의 사람이 갤러리에 포함되는가?



# 안면인식 개발의 주요 흐름

## PIAI Research Department

- **Face Localization (Detection+Cropping+ Alignment):** 이미지 내 얼굴 부분 감지 및 조정
- **Anti-spoofing:** 실제 얼굴과 모조된 얼굴(옷에 그려진 얼굴, 얼굴 그림 등) 구별
- **Face processing:** 안면에 다양한 각도에 대한 정면화 or 다양한 각도에 강인한 feature map 추출
- **Feature extraction :** 안면에서 중요한 feature map을 추출 (필요시 학습)
- **Face matching :** 추출된 feature map을 토대로 matching



- Verification : 두 얼굴이미지가 있을때, 같은 사람인지 다른 사람인지 판단(결과: Same / Different)
- Identification : 지정된 얼굴 ID 내에서 누구인지 판단 (결과: ID / nobody)
- Gallery : 인식하고자 하는 사람의 집합, ID와 사진으로 보통 구성됨
- Enroll : Gallery에 인식할 사람들을 등록하는 과정, ID와 Feature map을 매칭하여 gallery를 저장해놓음
- Bounding Box : 얼굴 검출 시에 얼굴위치를 알려주는 좌표 박스 (or bbox)
- Facial Landmark : 얼굴 내 눈, 코, 입 끝의 점의 위치를 알려주는 좌표 값들
- Feature map : 학습된 네트워크로 나온 결과 벡터값, 보통 주요 feature(우리는 알 수 없는 딥러닝이 학습한)의 값들의 모임(embedding layer, feature vector)
- Distance : 얼굴 Pair에서의 거리를 알려줄 수 있는 수치값 - 이를 토대로 verification, identification을 설정함
- Threshold : 검출이나 인식할때의 기준값, threshold 보다 높을 시 positive, 낮을시 negative로 인지





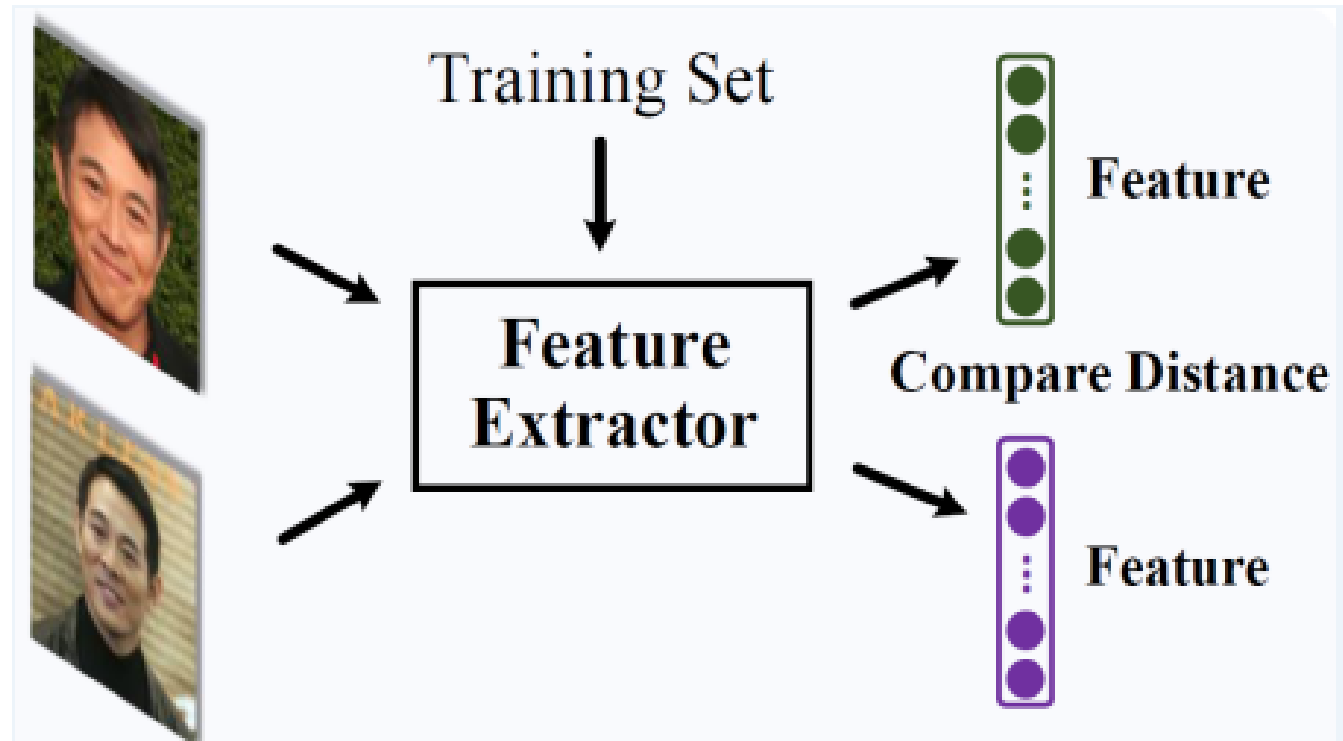
# Facial Verification On Image



# Facial Verification - Intro

PIAI Research Department

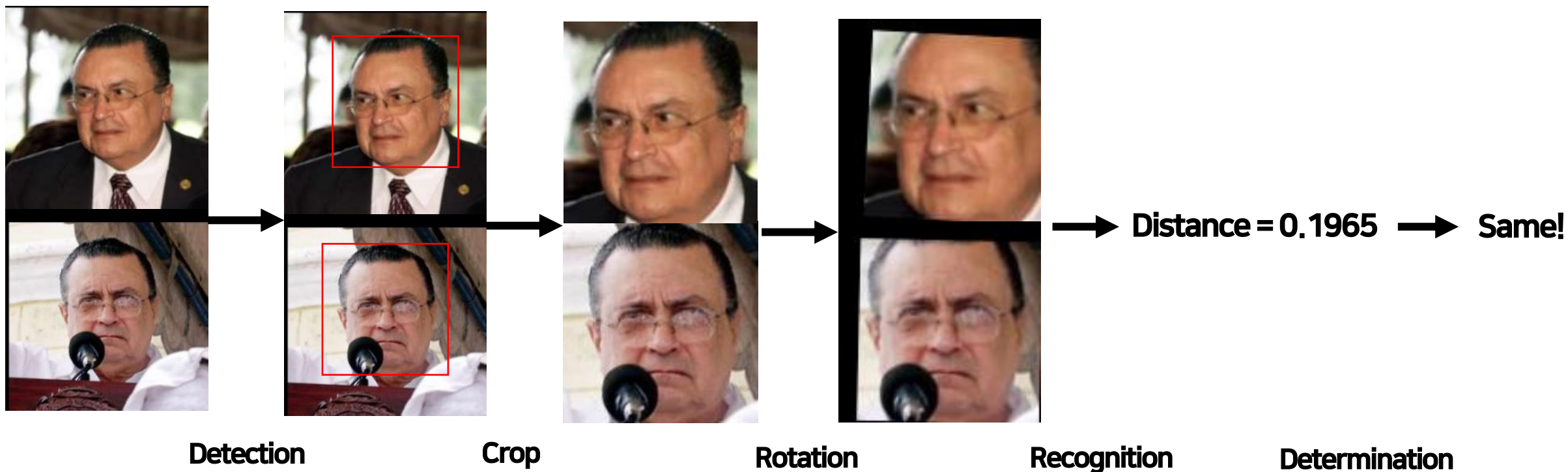
- Facial Verification : 두 쌍의 얼굴 이미지가 같은 사람인지, 다른 사람인지 판별하는 기술
- 같은 사람 : Positive Pair
- 다른 사람 : Negative Pair



# Facial Verification - 주요흐름

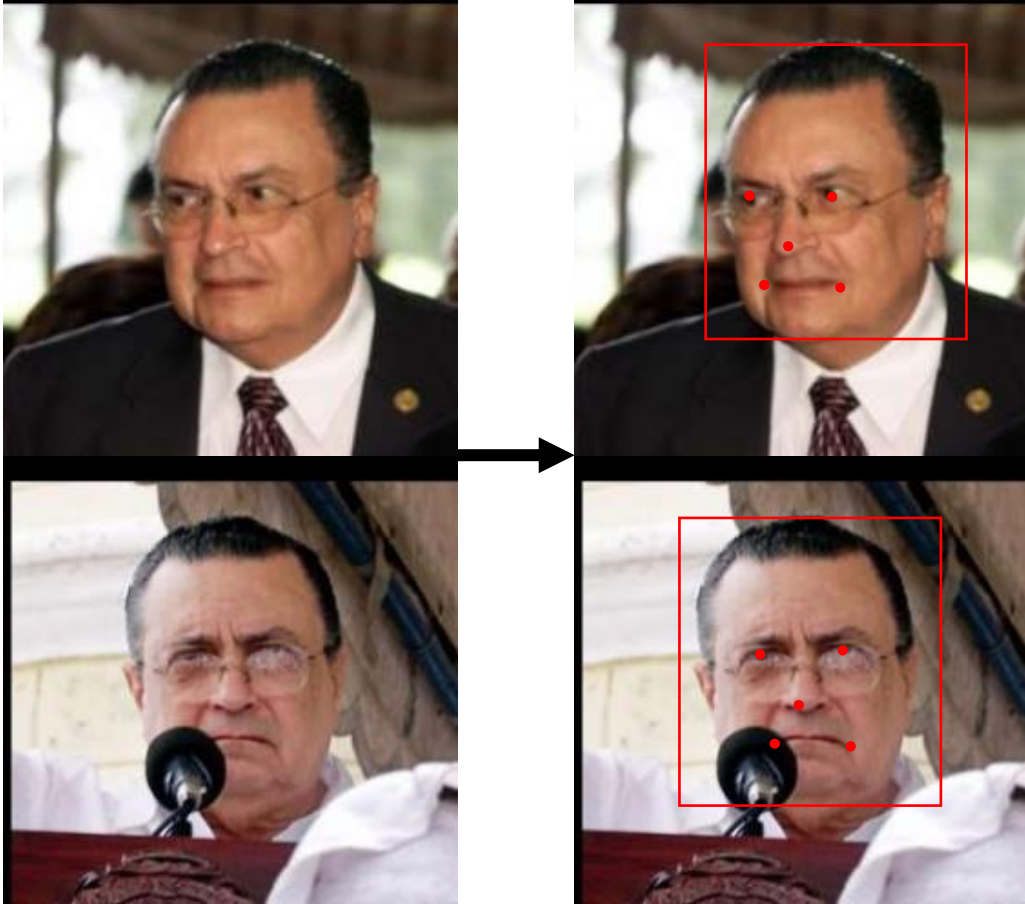
PIAI Research Department

- Facial Verification : 두 쌍의 얼굴이미지가 같은 사람인지, 다른 사람인지 판별하는 기술
- 같은 사람 : Positive Pair
- 다른 사람 : Negative Pair



# Facial Verification - Detection

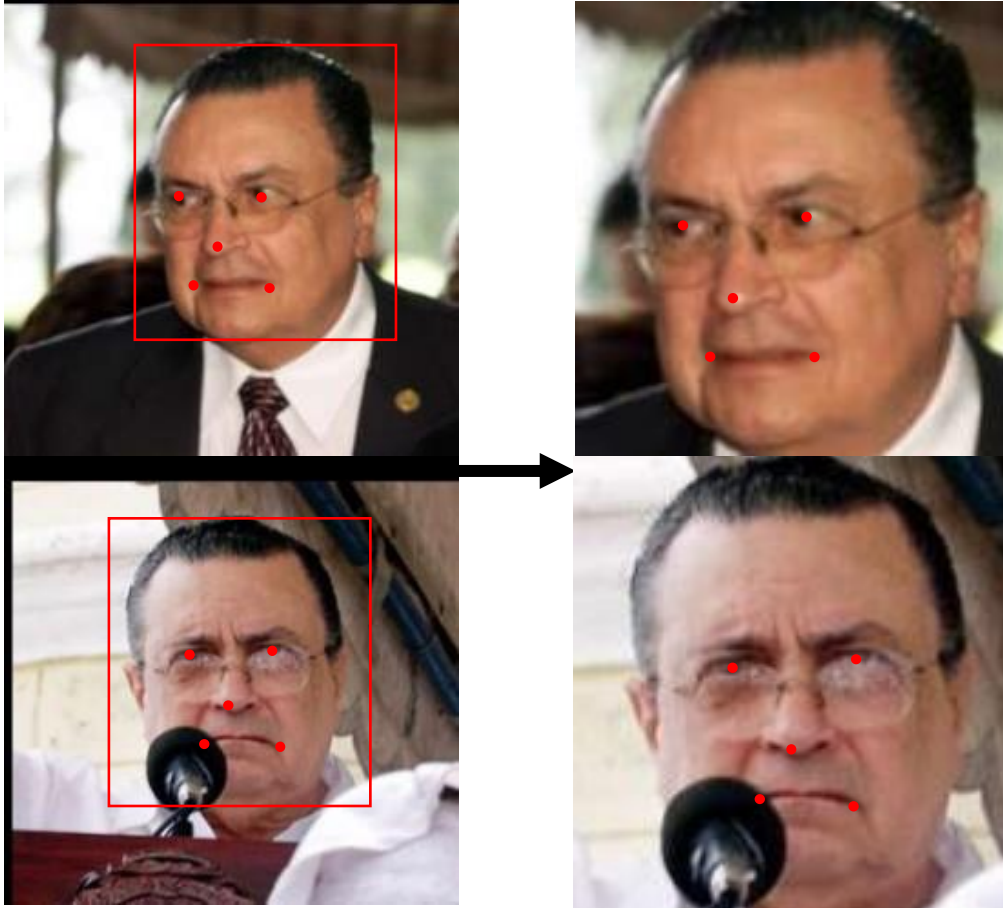
PIAI Research Department



- Facial Detection : 얼굴의 bounding box 좌표, facial landmark 좌표를 추출
- Face Recognition 전에 정사각으로 Cropping 하는 부분이 필요

# Facial Verification - Crop

PIAI Research Department



- Bounding Box: 얼굴 크기 형식상, 세로의 길이가 좀더 길게 detecting 되는 편
- 224\*224 정사각형태의 이미지를 만들기 위해 resize를 하게 될 시, 얼굴이 다소 납작해지는 현상 발생 가능
- 얼굴 형태를 유지하기 위해, bounding box의 중심좌표와 길이를 설정하고 이에 맞춰서 crop 진행
- 얼굴이 배경의 edge쪽에 있을 시, 오류 뜨지 않도록 선조치 필요

# Facial Verification - Rotation

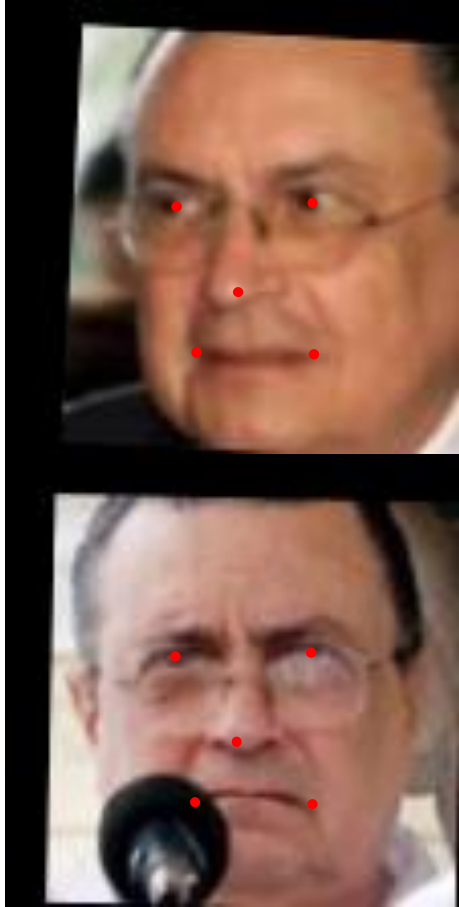
PIAI Research Department



- 얼굴 인식률을 높이기 위해서는 기본적인 눈코입 위치를 맞추는것이 좋다고 함.
- 전반적인 사람 얼굴의 평균적인 위치 좌표를 기반으로 rotation 수행
- Rotation에서 초과되는 부분은 검은 배경이 나오게끔 설정

# Facial Verification – Feature Distance

PIAI Research Department



Distance = 0.1965

- 두 얼굴이미지는 ResNet+Arcface Loss를 적용한 모델을 통해 feature map 추출(512 차원의 특징 벡터)
- Arcface 형식의 특징상 둘사이의 비교를 위한 feature map사이의  $\cos(\Theta)$ 거리 값 추출

$$distance = 1 - \frac{F_1 \cdot F_2}{|F_1||F_2|}$$

- 보통, distance가 0.68 이하이면 Positive pair, 아니면 negative pair일 확률이 큼



# Facial Verification - Detection

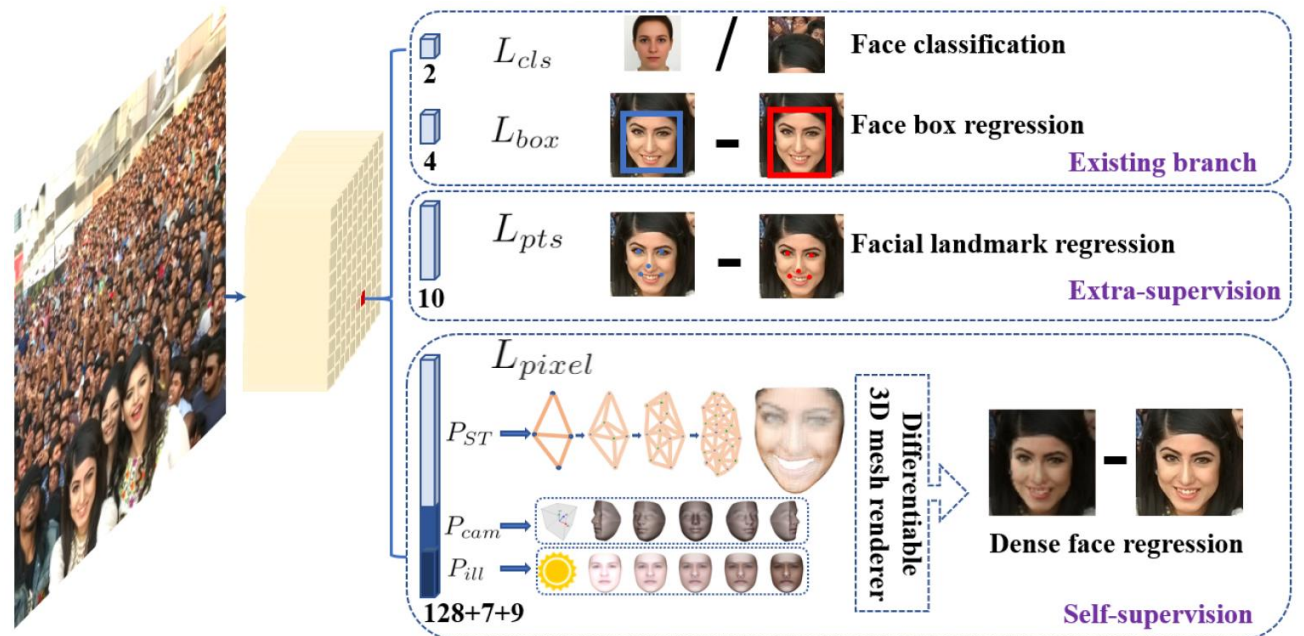
PIAI Research Department

## RetinaFace

### 1. Face localization 전체 작업을 multi-task로 training

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel}.$$

- $L_{cls}$  : Face classification loss
- $L_{box}$  : Face box regression loss
- $L_{pts}$  : Facial landmark regression loss
- $L_{pixel}$  : Dense regression loss
- $p_i$  : anchor  $i$ 의 face여부 predictive probability
- $p_i^*$  : positive anchor = 1, negative anchor = 0
- $t_i$  : predictive box position
- $t_i^*$  : GT box position
- $l_i$  : predictive 5 landmark position
- $l_i^*$  : GT 5 landmark position





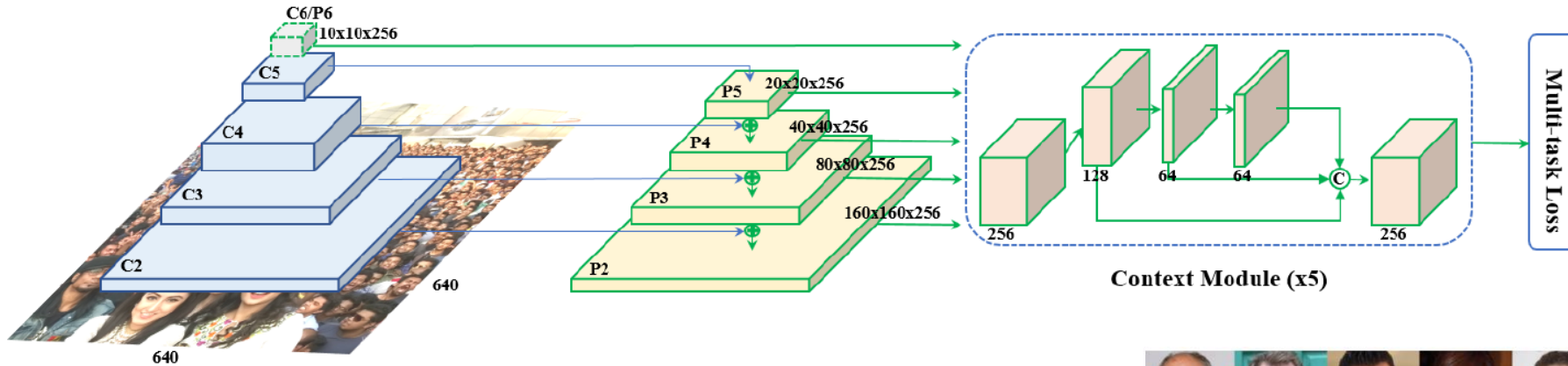
# Facial Verification - Detection

PIAI Research Department

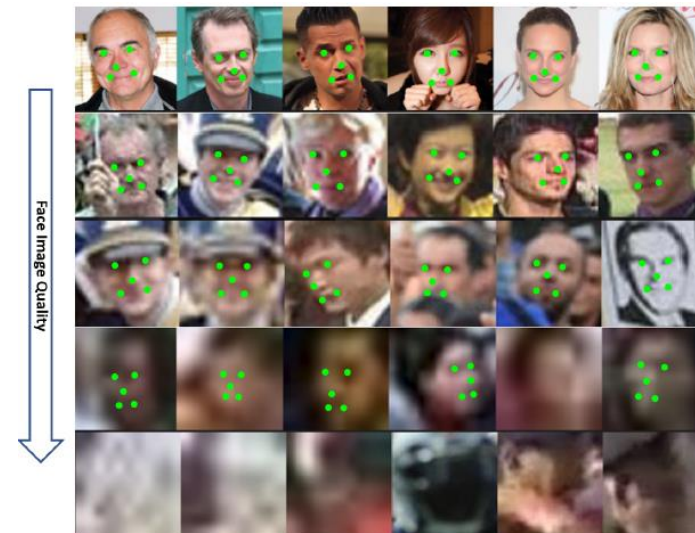
## RetinaFace

### 2. Improvement in hard face detection

- Feature Pyramid 방식을 통해 regression box 감지(SSH, Pyramid Box 착안)

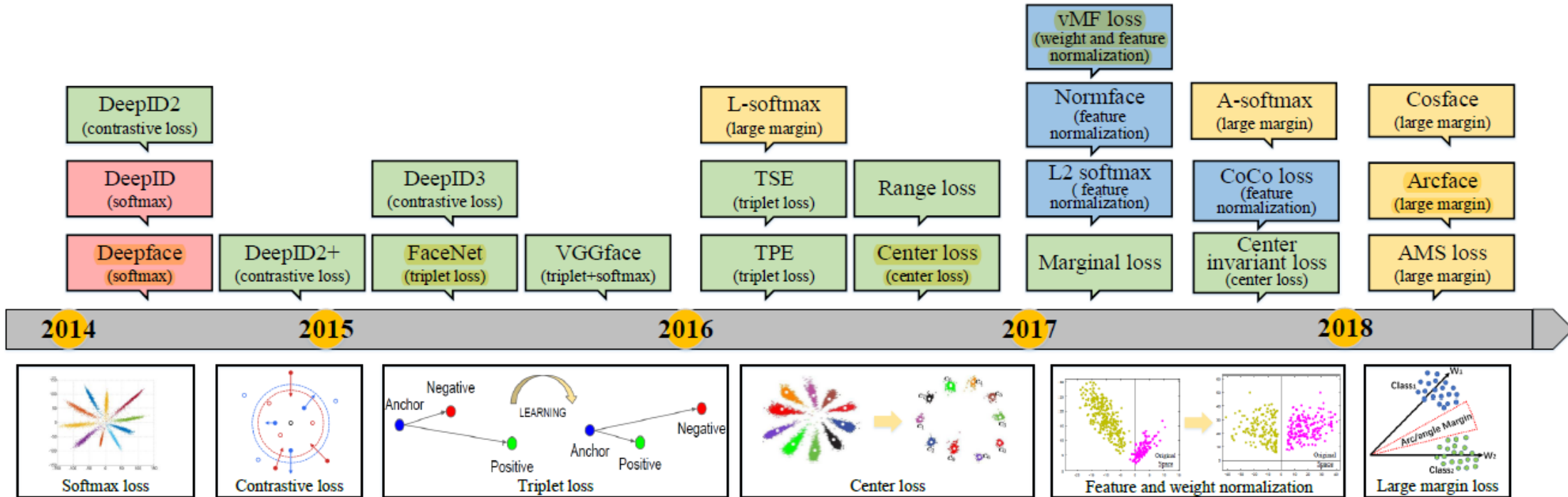


- 68 point facial landmark 대신, 5 point facial landmark 활용
- dataset에서의 facial landmark의 annotation level 설정,  
WIDER dataset 정제 작업 및 annotation 작업 수행 후 training



# Facial Verification – Recognition

PIAI Research Department



1. Softmax : 기존 image classification 전용 loss
  - Identity number가 많을수록, linear transformation matrix 크기가 커짐
  - Open set face recognition에 취약
2. Triplet Loss : 거리 개념을 통해 빠르게 train되도록 optimizing 해주는 loss
  - Large dataset의 경우 3쌍으로 구성하는 방식의 경우 수가 증가
  - 데이터의 질이 각 차이를 수치적으로 명시할 수 있을 만큼 좋아야 함
3. Center Loss : 같은 class에서의 center를 중심으로 모여지도록 optimizing 해주는 loss
  - Training 하는 face의 양이 많을수록, center update가 어려움
4. Normalization Loss : Feature, weigh를 평준화 시킨 후 구별시키는 loss
5. Angular/cosine-margin based Loss : 거리 개념을 Euclidean에서 극좌표계 형태로 바꾸는 loss

# Facial Verification – Recognition

PIAI Research Department

## ArcFace

### 1. Normalization + Angular Large Margin Loss의 양상별

#### Softmax Loss

$$L_1 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}}$$

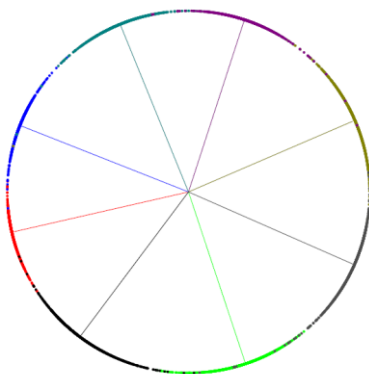
$x_i$  : deep feature of i-th sample

(실제 클래스:  $y_i$ )

$n$  : class number

$W$  : weight

$b$  : bias



(a) Softmax

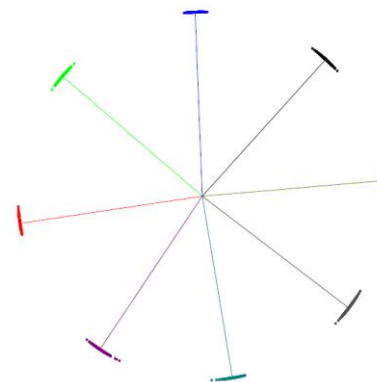
#### Arcface Loss

$$L_3 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$\theta_j$  :  $W_j$ 와  $x_i$ 사이의 angle  $W_j^T x_i = \|W_j\| \|x_i\| \cos \theta_j$

$m$  : additive angular margin penalty (geodesic distance margin penalty in the normalized hypersphere) (~0.5)

$s$  : hypersphere radius



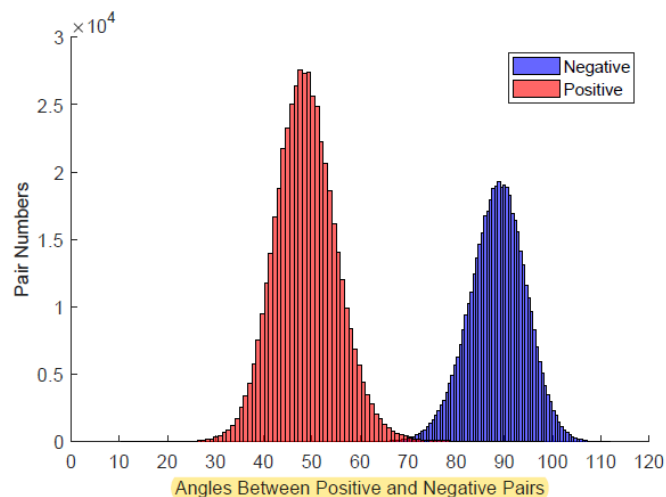
(b) ArcFace

# Facial Verification – Recognition

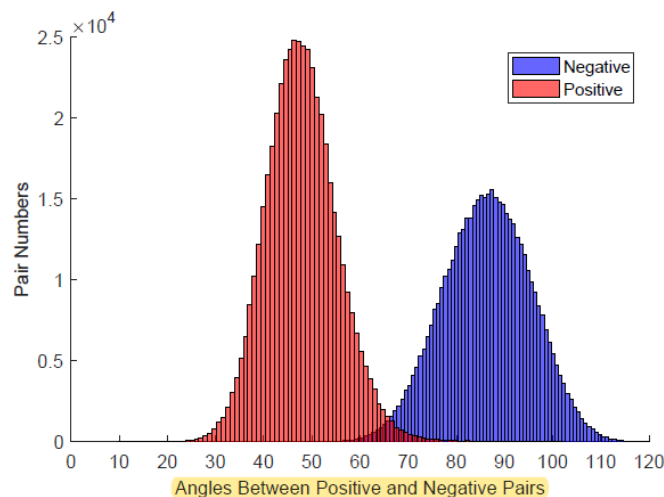
PIAI Research Department

## ArcFace

결과 비교



(a) ArcFace



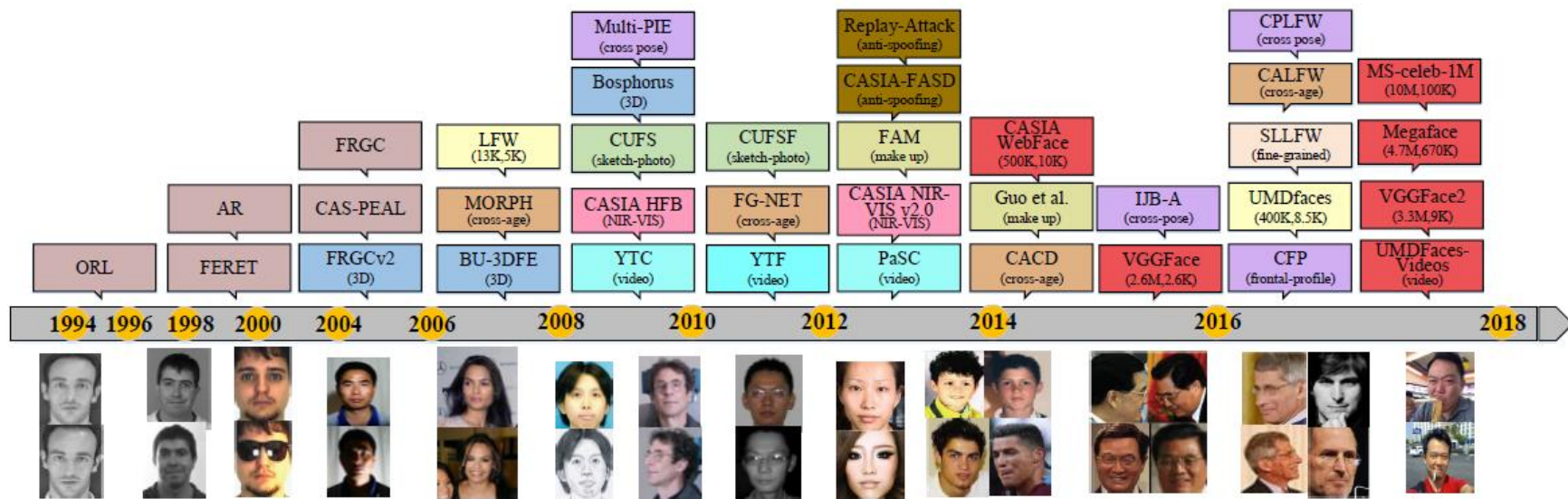
(b) Triplet-Loss

Method	#Image	LFW	YTF
DeepID [32]	0.2M	99.47	93.20
Deep Face [33]	4.4M	97.35	91.4
VGG Face [24]	2.6M	98.95	97.30
FaceNet [29]	200M	99.63	95.10
Baidu [16]	1.3M	99.13	-
Center Loss [38]	0.7M	99.28	94.9
Range Loss [46]	5M	99.52	93.70
Marginal Loss [9]	3.8M	99.48	95.98
SphereFace [18]	0.5M	99.42	95.0
SphereFace+ [17]	0.5M	99.47	-
CosFace [37]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	<b>99.83</b>	<b>98.02</b>

# Dataset

## PIAI Research Department

Datasets	Publish Time	#photos	#subjects	# of photos per subject <sup>1</sup>	Key Features
MS-Celeb-1M (Challenge 1) [69]	2016	10M 3.8M(clean)	100,000 85K(clean)	100	breadth; central part of long tail; celebrity; knowledge base
MS-Celeb-1M (Challenge 2) [69]	2016	1.5M(base set) 1K(novel set)	20K(base set) 1K(novel set)	1/-/100	low-shot learning; tailed data; celebrity
MS-Celeb-1M (Challenge 3) [2]	2018	4M(MSv1c) 2.8M(Asian-Celeb)	80K(MSv1c) 100K(Asian-Celeb)	-	breadth; central part of long tail; celebrity
MegaFace [105], [145]	2016	4.7M	672,057	3/7/2469	breadth; the whole long tail; commonality
VGGFace2 [22]	2017	3.31M	9,131	87/362.6/843	depth; head part of long tail; cross pose, age and ethnicity; celebrity
CASIA WebFace [243]	2014	494,414	10,575	2/46.8/804	celebrity
UMDFaces-Videos [10]	2017	22,075	3,107	-	video
VGGFace [149]	2015	2.6M	2,622	1,000	depth; celebrity; annotation with bounding boxes and coarse pose
CelebFaces+ [187]	2014	202,599	10,177	19.9	private
Google [176]	2015	>500M	>10M	50	private
Facebook [195]	2014	4.4M	4K	800/1100/1200	private





## LFW : Labeled Faces in the Wild

- 13,233 images, and 5749 사람으로 구성



- 모든 이미지가 300\*300 크기이며, 얼굴이 center로 나오도록 되어있음
- Data noise가 거의 없고 전반적인 data 질과 균일성이 특징
- 공인된 평가 데이터셋으로 쓰이는 편

테스트 : 3000 쌍의 동일인물, 3000쌍의 다른 인물로 두고 각각에 대한 distance 측정  
(이후 accuracy, ROC 값 추출로 비교)

# Dataset - FGLFW

PIAI Research Department

FGLFW : LFW에서 negative pair를 비슷한 인물로 설정한 testset  
- 다른 사람을 같은 사람으로 인식하는 문제에 강인한지 판별하기에 효과적

Negative pairs in LFW



Coarse-grained difference

Negative pairs in FGLFW



Fine-grained difference

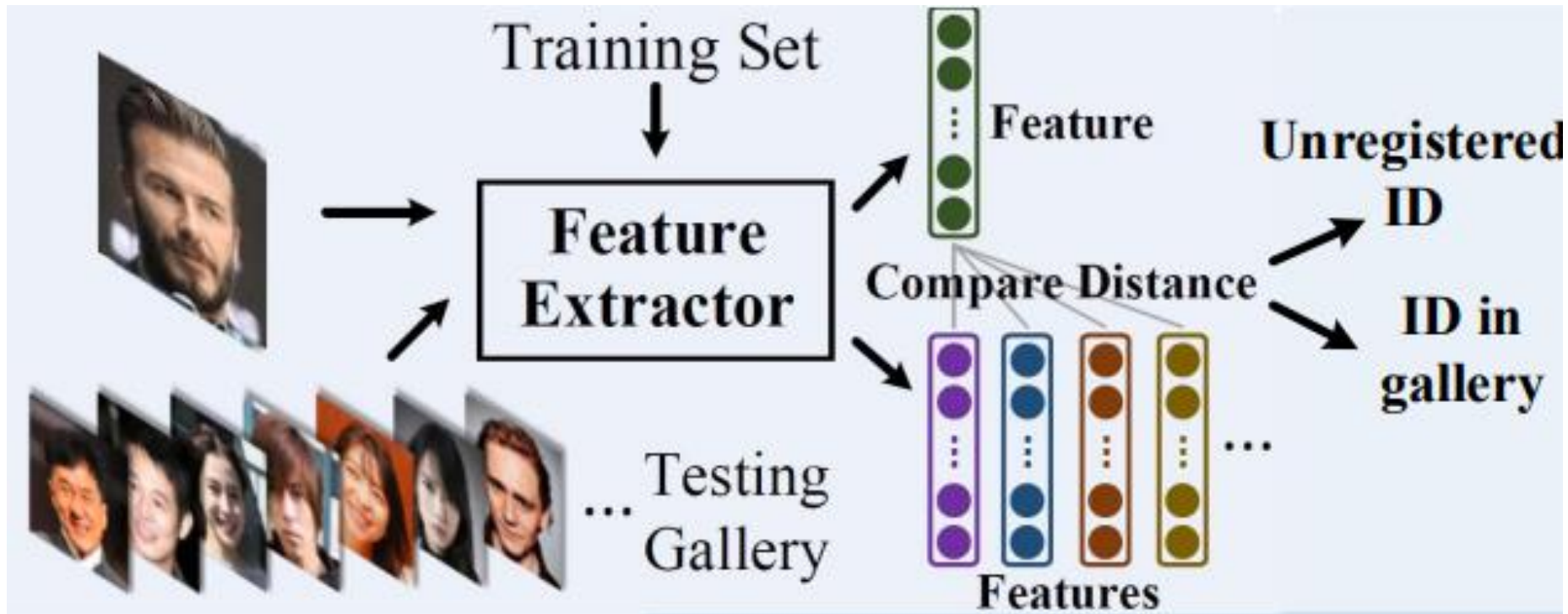


## Code Running

# Facial Identification - Intro

PIAI Research Department

- Facial Identification : Testing Gallery에 유사한 얼굴 이미지가 있는지를 판별, ID가 결과
- 흔하게 말하는 안면 식별 기술의 의미를 가지고 있음
- Gallery를 만드는 Enroll과 식별하고자 하는 Test로 코드를 나누어 진행



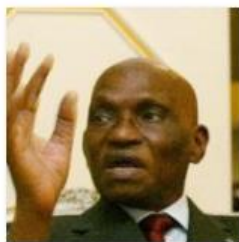
# Facial Identification - Enroll

## PIAI Research Department

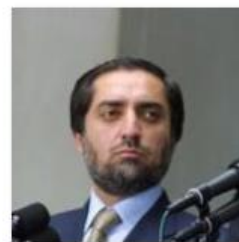
- Facial Enrollment : 테스트할 얼굴을 등록하는 과정
- 등록할 사람의 ID(보통은 이름이나, 동명이인이 있을 수 있는 경우 다른 형식으로)와 사진 이미지 필요
- 등록 기능에 따라서, 초기화(clear), 추가등록(enroll), 취소(delete), 조회(list)를 할 수 있어야함



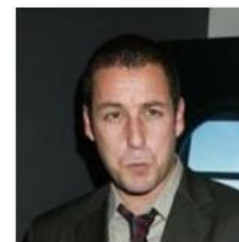
Abdel\_Nasser\_Assidi\_0001



Abdoulaye\_Wade\_0001



Abdullah\_0001



Adam\_Sandler\_0001

# Facial Enroll

PIAI Research Department

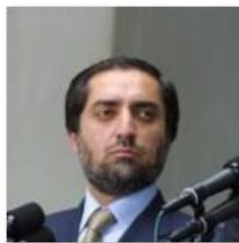
- 사진에서 Feature Map까지 추출하여 저장



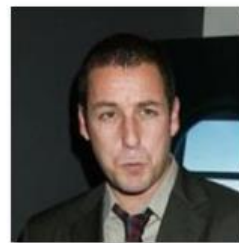
Abdel\_Nasser\_Assidi\_0001



Abdoulaye\_Wade\_0001



Abdullah\_0001



Adam\_Sandler\_0001

Detection/Crop/Rotation/Recognition

**Name**

**Feature Map**

[[Abdel_Nasser_Assidi	[[ 1.30243552e+00 3.74424815e-01 9.56335247e-01 3.62471968e-01 ...]]],
[Abdoulaye_Wade	[[ 9.39425945e-01 1.49660647e+00 -1.44220614e+00 -8.79160643e-01 ...]]],
[Abdullah	[[ 0.99086463 -0.90373546 -1.446125 0.5862401 1.4945618 -0.31188342 ...]]],
[Adam_Sandler	[[ -1.12069643e+00 9.05346692e-01 9.07279670e-01 5.06775141e-01 ...]]]

gallery.hdf에 저장

# Facial Identification - Determination

PIAI Research Department

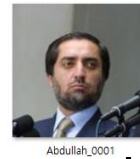
- 사진에서 Feature Map까지 추출하고, gallery의 feature map들 마다의 distance 추출
- 가장 높은 distance의 index에 있는 name 가장 유사한 얼굴의 ID
- Distance 값이 특정 Threshold를 넘을 시, 모든 gallery의 distance 가 높다는 뜻으로 stranger 확률 ↑



Name



Abdoulaye\_Wade\_0001



Abdullah\_0001



Adam\_Sandler\_0001

Feature Map

[[Abdel\_Nasser\_Assidi [[ 1.30243552e+00 3.74424815e-01 9.56335247e-01 3.62471968e-01 ...]],  
[Abdoulaye\_Wade [[ 9.39425945e-01 1.49660647e+00 -1.44220614e+00 -8.79160643e-01 ...]],  
[Abdullah [[ 0.99086463 -0.90373546 -1.446125 0.5862401 1.4945618 -0.31188342 ...]],  
[Adam\_Sandler [[ -1.12069643e+00 9.05346692e-01 9.07279670e-01 5.06775141e-01 ...]]]

Detection/Crop/Rotation/Recognition

[[[-1.12069643e+00 9.05346692e-01  
9.07279670e-01 5.06775141e-01 ...]]]



$$dist = 1 - \frac{F_1 \cdot F_2}{|F_1||F_2|}$$



[0.45, 0.51, 0.52, 0.51]

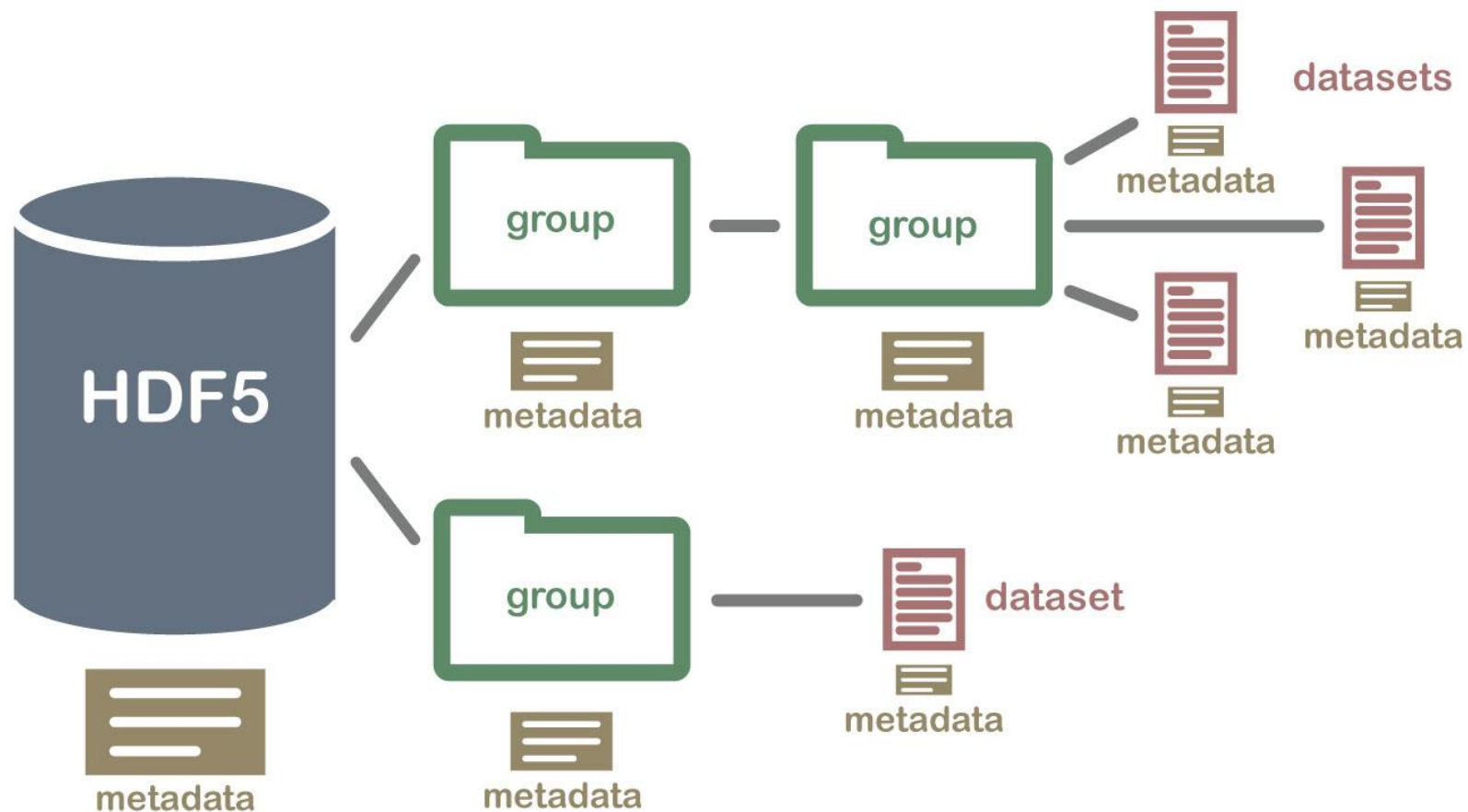


Stranger

# Facial Enroll

PIAI Research Department


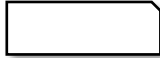
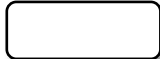
- HDF5 파일 구조

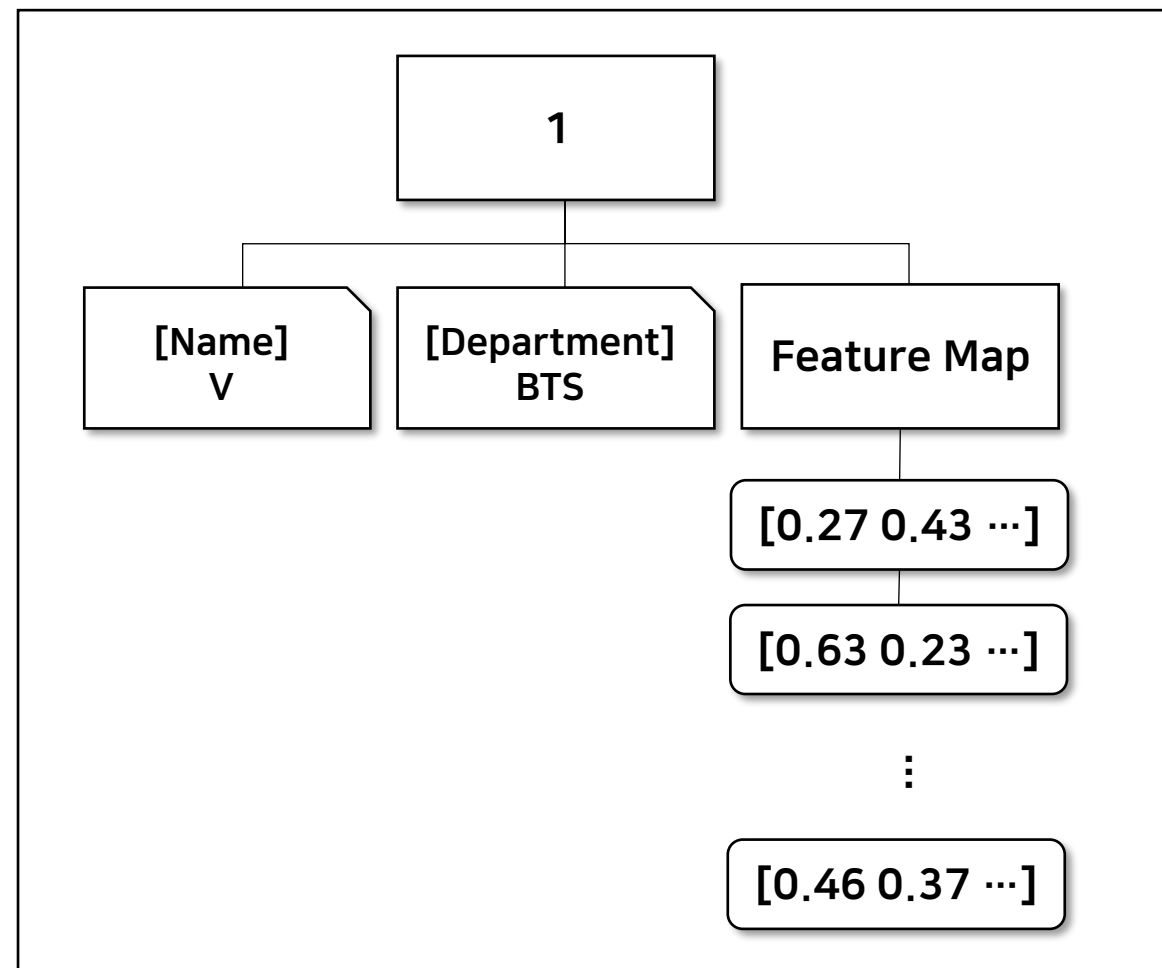
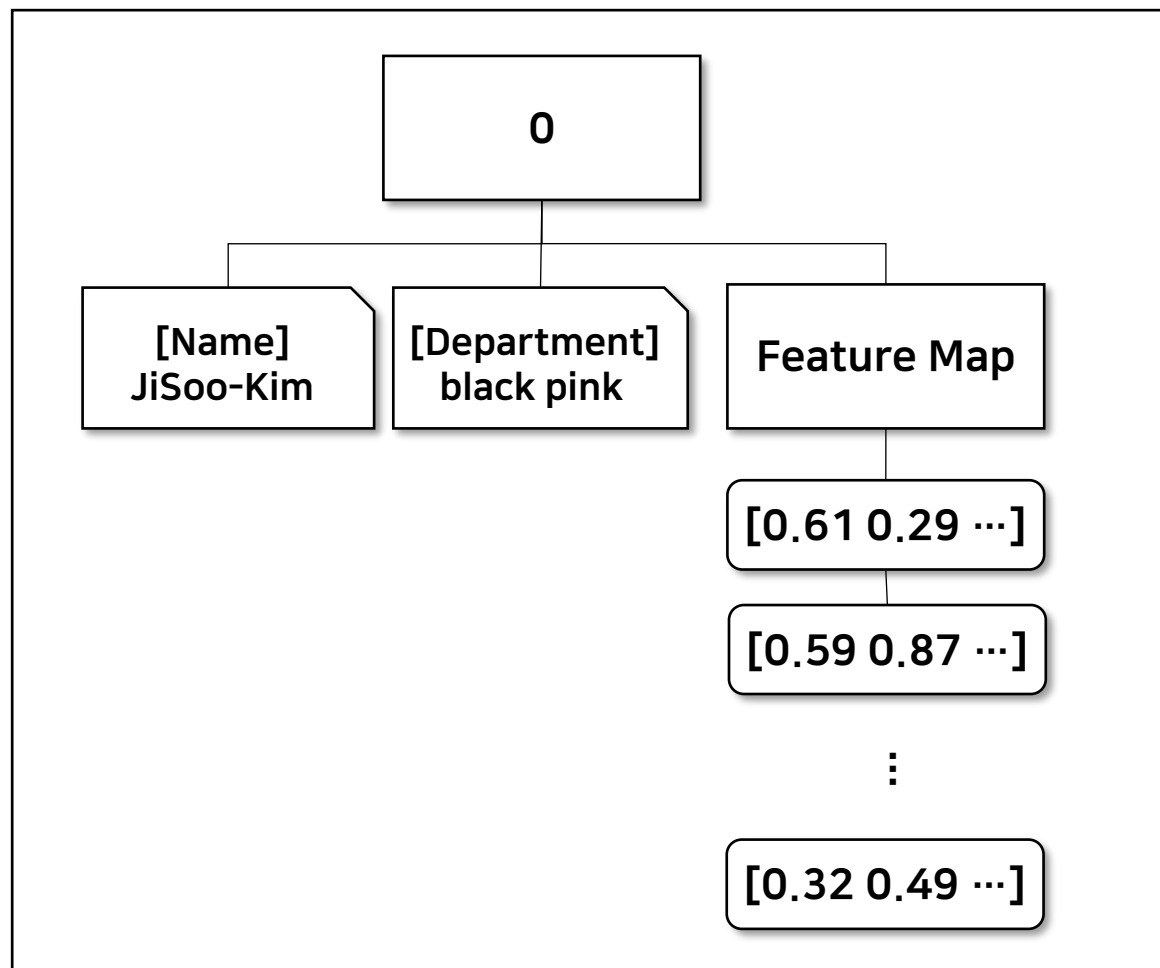


# Facial Enroll

PIAI Research Department

- HDF5로 구성한 Gallery

 : Group  
 : Meta Data  
 : Dataset





- HDF5 File Create

```
import h5py
```



h5py가 설치되어 있어야 함.

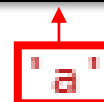
- conda install h5py
- pip install h5py

```
file_path = './data/recognition/gallery/'
```

```
file_name = 'gallery.hdf'
```

```
hdf_file = h5py.File(gallery_path + gallery_name, 'a')
```

- r : Read Only
- r+ : read/write
- w : create file / truncate if exists
- w- or x : create file / fail if exists
- a : read/write if exist, create if not exist



- HDF5 Create Group / Dataset

```
hdf_file.create_group('0')
```

```
hdf_file.create_dataset('Name', data='JiSoo-Kim')
```

```
hdf_file.create_dataset('Department', data='black pink')
```

```
hdf_file['0'].create_group('Feature Map')
```

```
hdf_file['0']['Feature Map'].create_dataset('feature_1', data=feature_map)
```

# Facial Identification Image

---

PIAI Research Department

# Code Running



# Metrics for Facial Verification & Identification

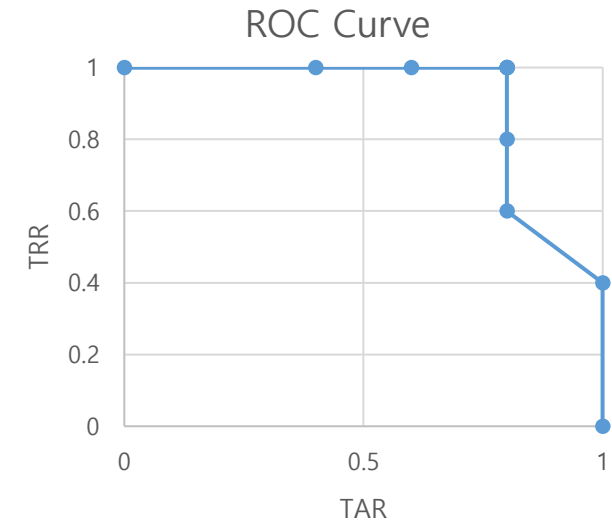
# Result Analysis - Verification

## PIAI Research Department

- Accuracy 와 AUC를 토대로 보통 정량적인 성과 체크
- TAR(True Acceptance Rate) : 실제- 같은 사람, 예측- 같은 사람 비율
- TRR(True Rejection Rate) : 실제- 다른 사람, 예측- 다른 사람 비율
- ROC(Receiver Operating Characteristic) : distance 결과에서 threshold 값에 따른 (TAR ,TRR) 그래프 곡선
- AUC(Area Under Curve) : ROC 아래쪽 면적 (AUC가 높을수록 우수한 인식률)

샘플	GT 값	distance
1	Same	0.99
2	Same	0.79
3	Same	0.89
4	Same	0.11
5	Same	0.92
6	Different	0.02
7	Different	0.11
8	Different	0.39
9	Different	0.21
10	Different	0.01

distance Threshold	TAR	TRR
1.0	0	1
0.9	0.4	1
0.8	0.6	1
0.7	0.8	1
0.6	0.8	1
0.5	0.8	1
0.4	0.8	1
0.3	0.8	0.8
0.2	0.8	0.6
0.1	1	0.4
0.0	1	0



- **AUC = 0.9**

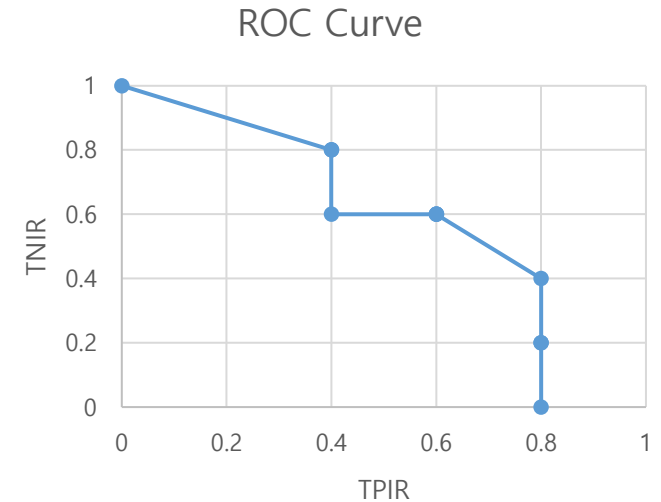
# Facial Identification - Analysis

## PIAI Research Department

- TPIR(True Positive Identification Rate) : 실제- 등록 사람, 예측- 동일 등록 사람 비율
- TNIR(True Negative Identification Rate) : 실제- 미등록 사람, 예측- 미등록 사람 비율
- ROC : distance 결과에서 threshold 값에 따른 (TPIR ,TNIR) 그래프 곡선
- AUC : ROC 아래쪽 면적 (AUC가 높을수록 우수한 인식률)

샘플	GT 값	결과
1	A	A, 0.99
2	B	B, 0.61
3	C	D, 0.71
4	D	D, 0.39
5	E	E, 0.98
6	미등록1	A, 0.22
7	미등록2	B, 0.31
8	미등록3	C, 0.99
9	미등록4	D, 0.71
10	미등록5	E, 0.01

distance Threshold	TPIR	TNIR
1.0	0	1
0.9	0.4	0.8
0.8	0.4	0.8
0.7	0.4	0.6
0.6	0.6	0.6
0.5	0.6	0.6
0.4	0.6	0.6
0.3	0.8	0.4
0.2	0.8	0.2
0.1	0.8	0.2
0.0	0.8	0



- ***AUC = 0.58***